Fine-Grained Classification with Noisy Labels

Qi Wei[1], Lei Feng[2], Haoliang Sun[1*], Ren Wang[1], Chenhui Guo[1], Yilong Yin[1*]
[1] School of Software, Shandong University, China
[2] School of Computer Science and Engineering, Nanyang Technological University, Singapore
{1998v7, haolsun.cn, xxlifelover, guiguigh1998}@gmail.com, feng0093@e.ntu.edu.sg
ylyin@sdu.edu.cn

Abstract

Learning with noisy labels (LNL) aims to ensure model generalization given a label-corrupted training set. In this work, we investigate a rarely studied scenario of LNL on fine-grained datasets (LNL-FG), which is more practical and challenging as large inter-class ambiguities among fine-grained classes cause more noisy labels. We empirically show that existing methods that work well for LNL fail to achieve satisfying performance for LNL-FG, arising the practical need of effective solutions for LNL-FG. To this end, we propose a novel framework called stochastic noise-tolerated supervised contrastive learning (SNSCL) that confronts label noise by encouraging distinguishable representation. Specifically, we design a noise-tolerated supervised contrastive learning loss that incorporates a weight-aware mechanism for noisy label correction and selectively updating momentum queue lists. By this mechanism, we mitigate the effects of noisy anchors and avoid inserting noisy labels into the momentum-updated queue. Besides, to avoid manually-defined augmentation strategies in contrastive learning, we propose an efficient stochastic module that samples feature embeddings from a generated distribution, which can also enhance the representation ability of deep models. SNSCL is general and compatible with prevailing robust LNL strategies to improve their performance for LNL-FG. Extensive experiments demonstrate the effectiveness of SNSCL.

1. Introduction

Learning from noisy labels [12, 13, 18, 21, 26, 40, 55, 58] poses great challenges for training deep models, whose performance heavily relies on large-scaled labeled datasets [28, 47–49]. Annotating data with high confidence would be resource-intensive, especially for some domains, such as medical and remote sensing images [29, 36, 37, 41, 46]. Thus, label noise would inevitably arise and then greatly degrade the generalization performance of deep models.

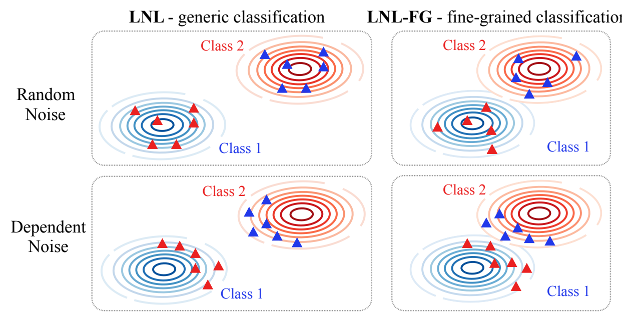

Figure 1. LNL-FG is more challenging than LNL on generic classification. ▲▲ denote mislabeled samples.

Previous methods [1, 6, 7, 9, 18, 23, 38, 53, 54] in LNL always focus on generic classification (*e.g.* CIFAR-10 & 100) and artificially construct random label noise [21, 23, 42, 43] and dependent label noise [9, 18, 38, 53, 55] to evaluate the performance of their algorithms. In this work, we extend LNL to *fine-grained* classification, which is a rarely studied task. Firstly, this scenario is more realistic since annotators are easier to be misguided by indistinguishable characteristics among fine-grained images and give an uncertain target. Fig. 1 illustrates comparison between two types of noise simulated on generic and fine-grained sets. Further, we extensively investigate the performance of prevailing LNL methods on our proposed LNL-FG task. The detailed results are shown in Fig. 2. Although these robust algorithms lead to statistically significant improvements over vanilla softmax cross-entropy on LNL, these gains do not transfer to LNL-FG task. Instead, some methods degrade the generalization performance of deep models compared to cross-entropy. Intuitively, due to large inter-class ambiguity among those classes in LNL-FG, the margin between noisy samples and the decision boundary in the fine-grained dataset is smaller than that in the generic dataset, leading to severe overfitting of deep models to noisy labels. Despite this fact, the typical method for better representation,

* denotes corresponding author

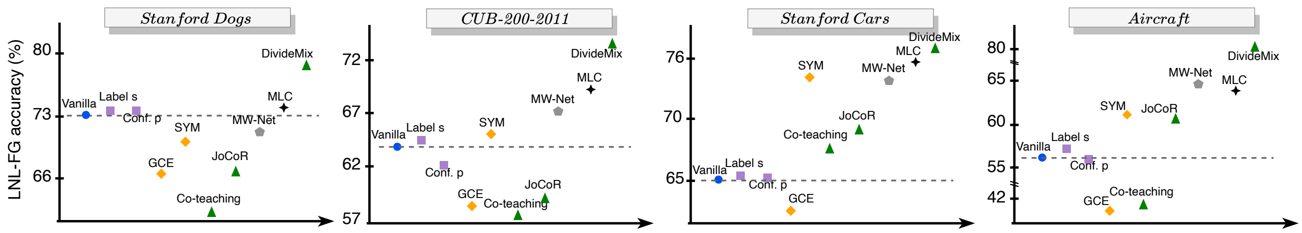

Figure 2. Comparison results of previous methods on four fine-grained benchmarks with 20% random label noise. Methods with same color and shape belong to the same strategy. The **X-axis** denotes their performance on typical LNL tasks while the **Y-axis** denotes that on LNL-FG tasks. It is obvious that **not all robust methods outperform the performance of *vanilla* cross-entropy on LNL-FG task.**. More analysis and results can be found in Appx. A.

i.e., DivideMix, consistently achieves better performance on both LNL and LNL-FG tasks (See Fig. 2). From this perspective, we consider that encouraging discrimitive feature not only confronts overfitting to label noise but also facilitates the learning of fine-grained task.

For this, contrastive learning (CL), as a powerful unsupervised learning approach for generating discrimitive feature [4,8,11,14,31], has attracted our attention. CL methods usually design objective functions as supervised learning to perform pretext similarity measurement tasks derived from an unlabeled dataset, which can learn effective visual representations in downstream tasks, especially for fine-grained classification [3]. The following work, supervised contrastive learning (SCL) [15], leverages label information to further enhance representation learning, which can avoid a vast training batch and reduce the memory cost. However, SCL cannot be directly applied to the noisy scenario as it is lack of noise-tolerated mechanism.

To resolve the noise-sensitivity of SCL, we propose a novel framework named stochastic noise-tolerated supervised contrastive learning (SNSCL), which contains a noise-tolerated contrastive loss and a stochastic module. For the noise-tolerated contrastive loss, we roughly categorize the noise-sensitive property of SCL into two parts of noisy anchors and noisy query keys in the momentum queue. To mitigate the negative effect introduced by noisy anchors or query keys, we design a weight mechanism for measuring the reliability score of each sample and give corresponding weight. Based on these weights, we modify the label of noisy anchors in current training batch and selectively update the momentum queue for decreasing the probability of noisy query keys. These operations are adaptive and can achieve a progressive learning process. Besides, to avoid manual adjustment of strong augmentation strategies for SCL, we propose a stochastic module for more complex feature transformation. In practice, this module generates the probabilistic distribution of feature embedding. By sampling operation, SNSCL achieves better generalization performance for LNL-FG.

Our contributions can be summarized as

- We consider a hardly studied LNL task, dubbed LNL-FG and conduct empirical investigation to show that some existing methods in LNL cannot achieve satisfying performance for LNL-FG.
- We design a novel framework dubbed stochastic noise-tolerated supervised contrastive learning (SNSCL), which alters the noisy labels for anchor samples and selectively updates the momentum queue, avoiding the effects of noisy labels on SCL.
- We design a stochastic module to avoid manually-defined augmentation, improving the performance of SNSCL on representation learning.
- Our proposed SNSCL is generally applicable to prevailing LNL methods and significantly improves their performance on LNL-FG.

Extensive experiments on four fine-grained datasets and two real-world datasets consistently demonstrate the state-of-the-art performance of SNSCL, and further analysis verify its effectiveness.

2. Related Work

Robust methods in Learning with noisy labels. The methods in the field of learning with noisy labels can be roughly categorized into robust loss function, sample selection, label correction, and sample reweight. The early works [22,25,52,60] mainly focus on designing robust loss functions which provide the deep model with greater generalization performance compared with the cross-entropy loss and contain the theoretical guarantee [22, 25]. Currently, more works turn to explore the application of the other three strategies. In label correction, researchers refurbish the noisy labels by self-prediction of the model's output [39, 51] or an extra meta-corrector [56, 61]. The latter enables admirable results of correction with a small set of meta-data. In sample section, the key point is how effective the preset selection criterion is. Previous literatures leverage the small-loss criterion that selects the examples with small empirical loss as the clean one [9, 53]. Recently, the works [1, 27, 55] represented by SELF [27] pay more attention to history prediction results, providing selection with more information and thus promoting the selection results. Besides, sample reweight methods [34,38] give examples with different weights, which can be regarded as a special form of sample selection. For example, [38] designed a meta-net for learning the mapping from loss to sample weight. The samples with large losses are seen as the noise, and thus meta-net generates small weights.

Contrastive learning. As an unsupervised learning strategy, contrastive learning [4, 5, 11] leverages similarity learning and markedly improves the performance of representation learning. The core idea of these methods is maximizing (minimizing) similarities of positive (negative) pairs at the data points.

CL has also been applied to LNL field for better representation learning and tackle negative effects of noisy labels. Sel-CL [20] proposes a pair-wise framework of selecting clean samples and conducts contrastive learning on those samples. Our proposed NTSCL is different in three aspects: 1) a different selection strategy via a novel weight-aware mechanism; 2) a stochastic module avoiding manually-defined augmentations in SCL for LNL. 3) a plug-and-play module for typical LNL methods. Our method NTSCL can be easily integrated into existing methods for improving performance on LNL or LNL-FG, while Sel-CL cannot. Besides, li *et al.* [19] introduces the ideas of momentum prototypes and trains the network such that embeddings are pulled closer to their corresponding prototypes, while pushed away from other prototypes. Due to the large inter-class ambiguity in fine-grained datasets, the quality of constructed class prototypes may be challenged.

3. Preliminaries

Problem definition. Assume $\mathcal{X}$ is the feature space from which the examples are drawn, and $\mathcal{Y} = \{1, 2, \cdots, C\}$ is the class label space, *i.e.* we consider a C-classification problem. Given a training set $\mathcal{D}^N = \{(\mathbf{x}_i, \mathrm{y}_i)\}_{i=1}^n$ with partial corrupted labels, where $(\mathbf{x}_i, \mathrm{y}_i)$ is drawn i.i.d. according to an distribution, over $(\mathcal{X}, \mathcal{Y})$. Supposing there is a deep classification network $F(\cdot)$ with the learnable parameters θ. For sample $\mathbf{x}$, the model's output can be written as $F(\mathbf{x}, \theta)$.

The goal of our algorithm is finding the optimal parameter θ^* which can achieve admirable generalization performance on the clean testing set.

Contrastive learning meets noisy labels. *Contrastive learning* [4, 8, 11] is a prevailing framework for representation learning, enhancing class discrimination of the feature extractor. Supposing a feature anchor q and a set of feature keys $\{\hat{\mathrm{q}}, \mathrm{k}_1, \cdots, \mathrm{k}_\mathrm{D}\}$ are given, where $\hat{\mathrm{q}}$ is a positive data point for q, and the others are negative. In CL, a widely used loss function for measuring the similarity of each data point is InfoNCE [30] and can be summarised as

$$\mathcal{L}_{\mathrm{INFO}} = -\log \frac{\exp\left(\mathrm{q}\cdot\hat{\mathrm{q}}/\tau\right)}{\exp\left(\mathrm{q}\cdot\hat{\mathrm{q}}/\tau\right) + \sum_{d=1}^{\mathrm{D}} \exp\left(\mathrm{q}\cdot\mathrm{k}_d/\tau\right)},$$

where τ is a hyper-parameter for temperature scaling. In most applications, CL is built as a pre-task. q and $\hat{\mathrm{q}}$ are extracted from two augmented views of the same example, and negative keys $\{\mathrm{k}_1, \cdots, \mathrm{k}_\mathrm{D}\}$ represent feature embeddings of other samples in the current training batch. CL is naturally independent of noisy labels, but there exists a drawback in that it lacks a mechanism to utilize potential labels into model training, leaving useful discriminative information on the shelf [50]. Currently, *supervised contrastive learning* [15] solves this issue by constructing the positive and the negative lists according to the labels. For anchor point q, the objective function can be written as

$$\mathcal{L}_{\mathrm{SCL}} = -\log \frac{\sum\limits_{\mathrm{k_P}\in\mathrm{Pos}} \exp\left(\mathrm{q}\cdot\mathrm{k_P}/\tau\right)}{\sum\limits_{\mathrm{k_P}\in\mathrm{Pos}} \exp\left(\mathrm{q}\cdot\mathrm{k_P}/\tau\right) + \sum\limits_{\mathrm{k_N}\in\mathrm{Neg}} \exp\left(\mathrm{q}\cdot\mathrm{k_N}/\tau\right)},$$

where Pos and Neg represent the positive and negative list, respectively.

However, SCL is sensitive to noisy labels, which can be introduced into the anchor point, Pos, and Neg. Our goal is to utilize the valuable information of the labels underlying the noisy training set $\mathcal{D}^N$ and overcome the misguidance of noisy labels.

4. Proposed method

Overview. In section 4.1, we first introduce a noise-tolerated supervised contrastive learning method that incorporates a weight-aware mechanism for measuring the reliability score of each example. Based on this mechanism, we dynamically alter the unreliable labels and selectively insert them into the momentum-updated queue, combating two noise-sensitive issues of SCL, respectively. Then, in section 4.2, we design a stochastic module for the transformation of feature embeddings, which samples from a generated probabilistic distribution. Eventually, we exhibit the total training objective in section 4.3.

4.1. Noise-tolerated supervised contrastive learning

Weight-aware mechanism. We aim to measure the reliability score of each sample in the training set $\mathcal{D}^N$ and generate the corresponding weight. For this, we use the *small-loss* criterion, a common strategy in LNL, and leverage a two-component GMM to generate this reliability score. Firstly, we evaluate the training set $\mathcal{D}^N$ after each training epoch. For clarity, we omit the epoch sequence and attain a list of empirical losses $\{l_i\}_{i=0}^n$ among all samples, where $l_i = L(F(\mathbf{x}_i;\theta), \mathrm{y}_i)$. Note that $L(\cdot)$ is the employed loss function. GMM fits to this list and gives the reliability score of the probability that the sample is clean. For sample x_i, the reliability score γ_i can be written as $\gamma_i = \mathrm{GMM}(l_i \,|\, \{l_i\}_{i=0}^n)$, where $\gamma_i \in [0, 1]$. Then, we design a function to dynamically adjust the weight for all training samples according to the reliability score. The

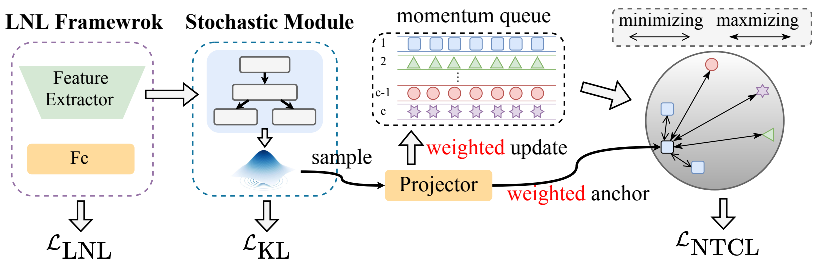

Figure 3. **Illustration of training framework**. Examples in the momentum queue with the same color and shape belong to the same category. The *Projector* is set as a single-layer MLP structure. Overall, the total training framework includes a LNL method and our proposed SNSCL, which consists of two parts: 1) **stochastic module**, which provides more competitive feature transformation for contrastive learning; 2) **noise-tolerated contrastive loss**, which is noise-aware and contains two weighting strategies.

weight of sample $\mathbf{x}_i$ is

$$\omega_i = \begin{cases} 1 & \text{if} \quad \gamma_i > t \\ \gamma_i & \text{otherwise} \end{cases}, \tag{1}$$

where t is a hyper-parameter in the interval of $[0, 1]$ and denotes the threshold of the reliability score. The computation of γ and ω restarts after each training round, ensuring that the values benefit from the improvement of the model's performance.

Based on this mechanism, we design two strategies that modify two noise-sensitive issues summarised in the overview. First, to solve the misguidance of the noisy anchor sample, we propose a **weighted correction strategy** to alter the labels of unreliable samples. For each sample $\mathbf{x} \in \mathcal{D}^N$, the weighted label $\hat{\mathrm{y}}$ is a soft label and is written as

$$\hat{\mathrm{y}} = (1-\omega)\mathrm{y}^{\mathrm{c}} + \omega \mathrm{y}, \tag{2}$$

where $\mathrm{y}^{\mathrm{c}} = \mathrm{Softmax}(F(x;\theta))$ and represents the prediction result of the classifier network. Indeed, this equation only change the labels of reliable samples, *i.e.*, $(\mathbf{x}, \mathrm{y}) \in \{(\mathbf{x}_i, \mathrm{y}_i) | \omega_i \neq 1\}_{i=1}^n$. Additionally, to make the alteration of labels more stable, we use the idea of moving-average. At epoch e, the moving-average corrected label over multiple training epochs is

$$\hat{\mathrm{y}}^e = \alpha \hat{\mathrm{y}}^{(e-1)} + (1-\alpha)\hat{\mathrm{y}}^e, \tag{3}$$

where $\alpha = 0.99$. Hence, the label-corrected training set can be formulated as $\{(\mathbf{x}_i, \hat{\mathrm{y}}_i^e)\}_{i=1}^n$ at e-th epoch . Note that all labels are represented via a soft one-hot vector.

Second, to solve the noise tolerance properties of the momentum queue, we propose a **weighted update strategy** to solve the noise-tolerant property of the traditional momentum queue. This strategy can be simply summarized as updating this queue according to the weight in Eq. 1. Given a sample $(\mathbf{x}_i, \hat{\mathrm{y}}_i)$, its weight value is ω_i. For the sample $\mathbf{x}_i$ which satisfies to $\omega_i = 1$, we update the y_i^h-th queue by its feature embedding via the First-in First-out principle, where y_i^h denotes the hard label and $\mathrm{y}_i^h = \arg\max(\hat{\mathrm{y}}_i)$. Otherwise, we update the y_i^h-th queue with probability ω_i. Intuitively, the weighted-update strategy avoids inserting unreliable samples into the queue, helping enhance the quality of the momentum queue.

4.2. Stochastic feature embedding

As reported in advanced works [11,45], typical CL heavily relies on sophisticated augmentation strategies and needs specify them for different datasets. We build a stochastic module to avoid manually-defined strategies. Given a sample x, let $\mathbf{z} = f(\mathbf{x})$ represent the output of the backbone network (*i.e.*, feature extractor) and $\mathbf{z} \in \mathbb{R}^d$. We formulate a probability distribution $p(Q|\mathbf{z})$ for embedding $\mathbf{z}$ as a normal distribution, which can be written as

$$p(Q|\mathbf{z}) \sim \mathcal{N}(\mu, \sigma^2), \tag{4}$$

where μ and σ can be learned by our stochastic module, a three-layers fully-connected network. From feature embedding distribution $p(Q|\mathbf{z})$, we sample an embedding $\mathbf{z}'$ to represent the augmented version of original feature embedding $\mathbf{z}$. Here, we use reparameterization trick [16],

$$\mathbf{z}' = \mu + \epsilon \cdot \sigma \quad \text{with} \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{5}$$

After that, the sampled feature embedding $\mathbf{z}'$ is utilized to update the momentum queue and compute contrastive learning loss. The merits of this module are 1) more complex representations are leveraged to stimulate the potential of CL, and 2) the property of stochasticity helps the model escape from memorizing the noisy signal to some degree. Module architecture has been discussed in Appx. C.1.

4.3. Total objective

We adopt online-update strategy that alternately trains the network, alters sample labels and updates the momentum queue. At e-th epoch, we have a label-corrected training set $\{(\mathbf{x}_i, \hat{\mathrm{y}}_i^e)\}_{i=1}^n$. For each sample in this set, the total training objective contains three parts.

Classification loss. Our proposal mitigates the effect of noisy labels by noise-tolerant representation learning, while a classification loss (*e.g.* cross-entropy) is required. Due to the flexibility, our framework can be easily integrated with prevailing LNL algorithms and leverages it for classifier learning. This loss item is written as $\mathcal{L}_{\mathrm{LNL}}$.

Noise-tolerated contrastive loss. For clarity, we omit the subscripts and formulate this sample as $(\mathbf{x}, \hat{\mathrm{y}})$ while q denotes its feature embedding and y^h represents its hard label. In our weighted momentum queue, the positive keys $\{\mathrm{k}_1^{\mathrm{y}^h}, \cdots, \mathrm{k}_D^{\mathrm{y}^h}\}$ are found according to the hard label y^h. Complementarily, the remaining key points in the momentum queue are regarded as negative keys with size $[\mathrm{D} \times (C-1)]$. Note that the size of the total momentum queue is $[\mathrm{D} \times C]$. Formally, our noise-tolerated contrastive loss is summarized as

$$
\begin{aligned}
\mathcal{L}_{\mathrm{NTCL}} &= -\frac{1}{\mathrm{D}} \sum\nolimits_{d=1}^{\mathrm{D}} \log \frac{\exp(\mathrm{q} \cdot \mathrm{k}_d^{\mathrm{y}^h}/\tau)}{L_{\mathrm{Pos}} + L_{\mathrm{Neg}}} \quad \text{with} \\
L_{\mathrm{Pos}} &= \sum\nolimits_{j=1}^{\mathrm{D}} \exp(\mathrm{q} \cdot \mathrm{k}_j^{\mathrm{y}^h}/\tau) \quad \text{and} \\
L_{\mathrm{Neg}} &= \sum\nolimits_{c=1}^{\{1,\cdots,C\}\backslash \mathrm{y}^h} \sum\nolimits_{j=1}^{\mathrm{D}} \exp(\mathrm{q} \cdot \mathrm{k}_j^{c}/\tau),
\end{aligned} \tag{6}
$$

where the L_{Pos} denotes positive keys from the same class y^h while L_{Neg} denotes the negative keys from other classes $\{1, \cdots, C\}\backslash \mathrm{y}^h$.

KL regularization. We employ the KL regularization term between the feature embedding distribution Q and unit Gaussian prior $\mathcal{N}(0, \mathbf{I})$ to prevent the predicted variance from collapsing to zero. The regularization can be formulated as

$$\mathcal{L}_{\mathrm{KL}} = \mathrm{KL}[p(\mathbf{z}|Q) || \mathcal{N}(0, \mathbf{I}))]. \tag{7}$$

The overall loss function can be formulated with two hyper-parameters λ_1 and λ_2 as

$$\mathcal{L} = \mathcal{L}_{\mathrm{LNL}} + \lambda_1 \mathcal{L}_{\mathrm{NTCL}} + \lambda_2 \mathcal{L}_{\mathrm{KL}}. \tag{8}$$

The training flowchart is shown in Fig. 3. Our proposed weighting strategies can be easily integrated into the typical SCL method, deriving a general LNL framework. The main operation is summarized in Algorithm 1. Compared to typical SCL, the weighting strategies would not cause much extra computational cost.

5. Experiments

5.1. Implementation details

Noisy test benchmarks. We introduce four typical datasets in fine-grained classification tasks and manually construct noisy labels. By a noise transition matrix T, we change partial labels of clean datasets. Given a noise ratio r, for a sample (x, y), the transition from clean label $y = i$ to wrong label $y = j$ can be represented by $\mathrm{T}_{ij} = \mathrm{P}(y = j | y = i)$ and $\mathrm{P} = r$, where r is the preset noise ratio. According to the structure of T, the noisy labels can be divided into two types: 1) **Symmetric** (random) noise. The diagonal elements of T are $1 - r$ and the off-diagonal values are $r/(c-1)$; 2) **Asymmetric** (dependent) noise. The diagonal elements of T are $1 - r$, and there exists another value r in each row. Noise ratio r is set as $r \in \{10\%, ..., 40\%\}$. Illustration of the matrix T is shown in Appx. B.1.

Algorithm 1 The training process of SNSCL

Require: Training set $\mathcal{D}^N$, a reliability threshold $t \in [0, 1]$, an average-moving coefficient α, two coefficients λ_1, λ_2.
Require: Classifier network $F(\theta)$, Stochastic module $\mathcal{M}$.
Ensure: Optimal parameters of classifier network θ^*

- WarmUp $(F(\theta); \mathcal{D}^N)$
- **while** $e <$ MaxEpoch **do**
  - Compute the loss and reliability score γ for each sample.
  - Compute the weight value ω for each sample. ▷ Eq. 1
  - Refurbish the labels with weighted-correct strategy and average-moving. ▷ Eq. 2, 3
  - **for** $iter \in \{1, ..., iters\}$ **do**
    - RrandomSample a batch $\{(\mathbf{x}_b, \mathrm{y}_b^e)\}_{b=1}^{\mathrm{B}}$ from the label-corrected training set, and compute loss $\mathcal{L}_{\mathrm{LNL}}$.
    - **for** $b \in \{1, ..., \mathrm{B}\}$ **do**
      - Sample feature embedding $\mathbf{z}'_b$ from the distribution $p(Q|\mathbf{z}_b)$. ▷ Eq. 4, 5
      - Weighted-update the momentum queue by $\mathbf{z}'_b$.
      - Compute two losses $\mathcal{L}_{\mathrm{KL}}$, $\mathcal{L}_{\mathrm{NTCL}}$. ▷ Eq. 6, 7
    - **end for**
    - Update$(\frac{1}{B} \sum\limits_{b=1}^{B} (\mathcal{L}_{\mathrm{LNL}} + \lambda_1 \mathcal{L}_{\mathrm{NTCL}} + \lambda_2 \mathcal{L}_{\mathrm{KL}}); \theta^{(e)})$.
  - **end for**
- **end while**
- **return** θ^*

We also select two noisy dataset collected from real world (*e.g.*, websites, crowdsourcing) to evaluate the effectiveness of our algorithm on real-world applications. 1) Clothing-1M [57] contains one million training images from 14 categories, with approximately 39.45% noisy labels. 2) Food-101N [2] contains 55k training images for 101 categories, with around 20% noise ratio.

Training settings. The code is implemented by Pytorch 1.9.0 with single GTX 3090. For four fine-grained noisy benchmarks, the optimizer is SGD with the momentum of 0.9, while initialized learning rate is 0.001 and the weight decay is 1e-3. The number of total training epochs is both 100, and the learning rate is decayed with the factor 10 by 20 and 40 epoch. For Clothing-1M, refers to [55], we train the classifier network for 15 epochs and use SGD with 0.9 momentum, weight decay of 5e-4. The learning rate is set as 0.002 and decayed with the factor of 10 after 10 epochs, while warm up stage is one epoch. For Food-101N, we train the classifier network for 50 epochs and use SGD with 0.9 momentum, weight decay of 5e-4. The learning rate is set as 0.002 and decayed with the factor of 10 after 30 epochs,

Table 1. Comparisons with test accuracy on **symmetric** label noise. The average **best** and the **last** accuracy among three times are reported. ↑ denotes the performance improvement of *SNSCL*.

	Stanford Dogs		Standford Cars		Aircraft		CUB-200-2011	
	20%	40%	20%	40%	20%	40%	20%	40%
Cross-Entropy	73.01 (63.82)	69.20 (50.45)	65.74 (64.08)	51.42 (45.62)	56.51 (54.67)	45.67 (38.89)	64.01 (60.77)	54.14 (45.85)
+ SNSCL	76.33 (75.83)	75.27 (75.00)	83.24 (82.99)	76.72 (76.36)	76.45 (76.45)	70.48 (69.64)	73.32 (72.99)	68.83 (68.67)
Label Smooth [24]	73.51 (64.42)	70.22 (50.97)	65.45 (64.24)	51.57 (45.19)	58.21 (54.73)	45.24 (38.01)	64.76 (60.60)	54.39 (45.28)
+ SNSCL	76.85 (76.12)	74.64 (74.60)	83.21 (83.01)	76.07 (75.90)	76.24 (75.70)	70.36 (70.06)	73.46 (73.09)	69.14 (68.64)
Conf. Penalty [33]	73.22 (66.89)	68.69 (52.98)	64.74 (64.46)	48.15 (43.71)	56.32 (55.51)	43.64 (39.54)	62.75 (61.10)	52.04 (45.13)
+ SNSCL	76.14 (75.73)	74.72 (74.49)	83.07 (83.00)	75.67 (75.38)	75.04 (74.23)	67.99 (66.85)	73.90 (73.51)	68.42 (67.86)
GCE [60]	66.96 (66.93)	61.47 (60.32)	62.77 (61.23)	47.44 (46.13)	39.54 (39.24)	32.34 (32.28)	58.74 (57.20)	49.71 (48.11)
+ SNSCL	75.99 (74.56)	71.68 (70.62)	73.78 (73.55)	58.11 (57.41)	72.67 (71.53)	60.19 (59.83)	70.83 (70.56)	61.67 (61.46)
SYM [52]	69.20 (62.13)	65.76 (46.99)	74.65 (73.21)	52.83 (51.61)	62.29 (60.51)	54.36 (45.39)	65.34 (63.60)	50.19 (50.15)
+ SNSCL	77.55 (77.24)	76.28 (76.25)	84.59 (83.54)	79.07 (78.87)	79.64 (79.09)	74.02 (73.63)	76.67 (76.06)	72.71 (72.58)
Co-teaching [9]	63.71 (58.43)	49.15 (48.92)	68.60 (67.95)	56.92 (55.95)	42.55 (40.62)	35.21 (32.16)	57.84 (55.98)	46.57 (46.22)
+ SNSCL	74.18 (73.09)	60.71 (58.84)	78.94 (78.13)	75.98 (75.06)	74.61 (74.19)	65.47 (63.81)	69.77 (69.34)	60.59 (58.94)
JoCoR [53]	66.94 (60.81)	49.62 (48.62)	69.99 (68.25)	57.95 (56.71)	61.37 (59.16)	52.11 (49.93)	58.79 (57.74)	52.64 (49.35)
+ SNSCL	75.79 (74.99)	63.42 (62.84)	79.67 (78.77)	76.80 (76.21)	75.88 (75.16)	71.65 (70.67)	71.86 (70.90)	64.43 (63.81)
MW-Net [38]	71.99 (69.20)	68.14 (65.17)	74.01 (73.88)	58.30 (55.81)	64.97 (61.84)	57.61 (55.90)	67.44 (65.20)	58.49 (54.81)
+ SNSCL	77.49 (77.08)	74.92 (74.38)	85.96 (85.37)	77.76 (77.13)	80.08 (78.94)	73.55 (73.18)	76.94 (76.24)	69.51 (68.83)
MLC [61]	74.08 (70.51)	69.44 (66.28)	76.02 (71.24)	59.44 (55.76)	63.81 (60.33)	58.11 (54.86)	69.44 (68.19)	60.27 (58.49)
+ SNSCL	78.92 (78.56)	76.49 (78.96)	85.92 (84.91)	78.49 (77.80)	79.19 (78.40)	75.21 (74.67)	77.58 (76.68)	71.54 (70.86)
DivideMix [18]	79.22 (77.86)	77.93 (76.28)	78.35 (77.99)	62.54 (62.50)	80.62 (80.50)	66.76 (66.13)	75.11 (74.54)	67.35 (66.96)
+ SNSCL	81.40 (81.16)	79.12 (78.91)	86.29 (85.94)	80.09 (79.51)	82.31 (82.03)	76.22 (75.67)	78.36 (78.04)	73.66 (73.28)
Avg. ↑	**5.88 (9.34)**	**7.76 (15.83)**	**12.44 (13.29)**	**20.82 (23.06)**	**18.60 (19.86)**	**21.41 (24.49)**	**9.87 (11.25)**	**12.22 (16.46)**

Table 2. Comparisons with test accuracy on **asymmetric** label noise. The average **best** and the **last** accuracy among three times are reported. ↑ denotes the performance improvement of *SNSCL*.

	Stanford Dogs		Standford Cars		Aircraft		CUB-200-2011	
	10%	30%	10%	30%	10%	30%	10%	30%
Cross-Entropy	74.24 (71.32)	63.76 (56.86)	74.58 (74.57)	58.08 (57.43)	65.98 (62.53)	51.10 (47.85)	68.26 (68.00)	56.02 (54.13)
+ SNSCL	76.24 (74.88)	64.49 (62.37)	83.73 (83.41)	70.04 (69.61)	78.28 (78.22)	65.44 (65.11)	74.80 (74.47)	61.48 (60.70)
Label Smooth [24]	74.70 (71.81)	64.99 (57.04)	74.28 (74.13)	58.47 (57.80)	65.29 (63.34)	51.88 (47.71)	68.78 (67.67)	56.80 (53.69)
+ SNSCL	75.84 (75.16)	65.23 (63.69)	84.27 (84.13)	70.49 (70.20)	78.67 (77.98)	66.28 (65.56)	75.51 (75.42)	62.05 (61.43)
Conf. Penalty [33]	74.41 (72.04)	64.50 (57.92)	73.78 (73.67)	56.96 (56.53)	64.90 (63.01)	49.38 (47.53)	67.66 (67.62)	54.33 (52.80)
+ SNSCL	76.01 (75.62)	67.53 (66.32)	84.26 (83.91)	72.23 (71.96)	78.34 (78.01)	66.88 (66.34)	75.34 (74.97)	62.69 (62.67)
GCE [60]	67.13 (66.83)	54.53 (53.92)	68.75 (68.71)	60.57 (60.21)	44.22 (44.16)	34.18 (33.66)	62.92 (60.77)	50.05 (49.79)
+ SNSCL	75.91 (74.63)	68.45 (67.13)	80.33 (80.04)	64.64 (64.38)	73.85 (73.89)	64.33 (63.91)	73.77 (73.23)	61.37 (60.96)
SYM [52]	69.57 (66.75)	61.61 (51.11)	76.74 (76.18)	58.30 (57.42)	69.31 (67.45)	50.23 (47.55)	68.81 (68.00)	52.16 (51.83)
+ SNSCL	77.37 (76.64)	74.74 (74.41)	86.71 (86.54)	78.98 (78.66)	82.30 (81.46)	69.61 (69.37)	77.89 (77.27)	67.43 (66.95)
Co-teaching [9]	59.95 (59.77)	50.50 (50.44)	72.88 (72.71)	61.02 (60.86)	55.94 (49.85)	45.18 (38.97)	61.00 (60.92)	50.06 (48.55)
+ SNSCL	70.46 (70.24)	65.83 (65.41)	82.17 (81.63)	66.84 (66.49)	74.73 (74.28)	62.17 (61.88)	70.92 (70.63)	64.55 (64.10)
JoCoR [53]	61.34 (60.11)	53.39 (52.35)	74.68 (73.21)	63.54 (62.27)	67.12 (64.99)	52.25 (50.28)	62.99 (61.88)	51.70 (49.60)
+ SNSCL	74.26 (72.96)	70.40 (70.01)	83.67 (83.28)	71.74 (71.22)	78.84 (78.29)	67.50 (66.48)	74.52 (73.97)	66.07 (65.26)
MW-Net [38]	73.68 (72.19)	65.81 (65.19)	76.27 (75.89)	65.19 (63.32)	72.76 (70.18)	54.88 (51.80)	67.44 (65.08)	57.49 (56.10)
+ SNSCL	78.52 (78.03)	72.68 (72.20)	85.73 (85.44)	75.69 (75.28)	80.69 (80.22)	70.49 (69.90)	76.07 (76.70)	68.95 (68.26)
MLC [61]	75.84 (74.99)	69.81 (69.03)	77.80 (77.29)	67.93 (67.28)	74.40 (73.91)	59.44 (59.00)	68.84 (68.21)	58.73 (58.29)
+ SNSCL	79.22 (78.96)	75.92 (75.57)	87.05 (86.70)	79.44 (79.21)	82.75 (82.43)	72.30 (71.96)	76.91 (76.47)	69.70 (69.24)
DivideMix [18]	79.39 (78.47)	75.51 (73.67)	79.34 (77.92)	68.69 (68.63)	76.57 (76.24)	63.97 (63.28)	72.76 (71.24)	63.65 (62.68)
+ SNSCL	81.90 (81.72)	77.19 (77.02)	88.18 (87.94)	81.44 (80.96)	84.17 (84.03)	74.80 (74.57)	78.92 (78.56)	71.28 (70.83)
Avg. ↑	**5.55 (6.57)**	**7.81 (10.6)**	**9.70 (9.87)**	**11.28 (11.62)**	**13.61 (15.31)**	**16.73 (18.74)**	**8.51 (9.23)**	**10.46 (11.30)**

while warm up stage is five epoch. For all experiments, we set the training batch size as 32. In addition, we adopt a default temperature $\tau = 0.07$ for scaling. More detailed setting, including augmentation strategies and applied backbone, can be found in Appx. B.2.

Hyper-parameters settings. Our framework SNSCL mainly includes two hyper-parameters, *i.e.*, the reliability threshold t in Eq. 1 and the length of momentum queue D Eq. 6. For all experiments, we set $t = 0.5$ and $\mathrm{D} = 32$. In addition, the trade-off parameters in Eq. 8 are set as $\lambda_1 = 1, \lambda_2 = 0.001$.

5.2. Comparison with state-of-the-arts

Baselines. We evaluate the effectiveness of our method by adding the proposal into current LNL algorithm and compare the improvements on LNL-FG task. The basic methods we compared include CE, Label Smooth [24], Confidence Penalty [33], Co-teaching [9], JoCoR [53], DivideMix [18], SYM [52], GCE [60], MW-Net [38], and MLC [61]. Settings about these methods are shown in Appx. B.3.

Results on four fine-grained benchmarks. We compare 10 algorithms and attain significant improvement of top-1 testing accuracy (%) on four fine-grained benchmarks. We

Table 3. Comparisons with test accuracy (%) on real-world benchmarks, including Clothing-1M (the *left*) and Food-101N (the *right*). The **solid** results denote the improvement of our method SNSCL. The average results among five times are reported.

Clothing-1M ($r \approx 39.5\%$)				Food-101N ($r \approx 20\%$)			
Forward [32]	69.84	SFT+ [55]	75.08	CleanNet [17]	83.47	WarPI [40]	85.91
JoCoR [53]	70.30	CE	64.54	MWNet [38]	84.72	CE	81.67
Joint Optim [42]	72.23	CE + SNSCL	**73.49**	NRank [35]	85.20	CE+SNSCL	**85.44**
SL [52]	71.02	DivideMix [18]	74.76	SMP [10]	85.11	DivideMix [18]	85.88
ELR+ [21]	74.81	DivideMix + SNSCL	**75.31**	PLC [59]	85.28	DivideMix+SNSCL	**86.40**

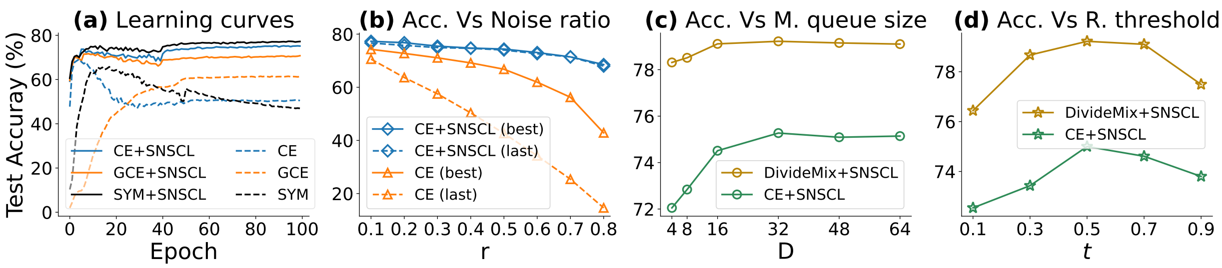

Figure 4. More analyses on Stanford dogs with 40% symmetric label noise from four perspectives .

show the results in Tab. 1 & 2, where we test symmetric and asymmetric noise types. To demonstrate the effectiveness of our method, we give experimental comparisons from two aspects. 1) **Improvements on top-1 test accuracy**. Overall, our method SNSCL achieves consistent improvement in all noisy conditions. The average minimal improvement is 5.55% in Stanford Dogs with 10% asymmetric noise, and the maximum is 21.41% in Aircraft with 40% symmetric noise. 2) **Mitigating overfitting on fine-grained sets**. In these tables, we report the best accuracy and the last epoch's accuracy. It is noteworthy that the investigated methods mainly overfit on these benchmarks (*i.e.*, the accuracy attaches a peak and then drops gradually, causing a great gap between these two values). However, SNSCL mitigates overfitting and maintains more stable learning curves.

Results on real-world noisy datasets. To evaluate the effectiveness of SNSCL on real-world applications, we conduct experiments on two datasets which are collected from websites. 1) **Clothing-1M**, the comparison results are shown in Tab. 3. We select cross-entropy, GCE and DivideMix as the basic methods and integrate SNSCL with them. Obviously, the combination *DivideMix+SNSCL* outperforms the state-of-the-art method SFT+ by 0.23% top-1 test accuracy. Moreover, in contrast with the bases, SNSCL achieves remarkable improvements by 8.95% and 0.55%, respectively. 2) **Food-101N**, the comparison results is shown in Tab. 3. Compared to basic methods, SNSCL brings significantly improvement by 3.77% and 0.52%, respectively. These results demonstrate the effectiveness of our methods in real-world applications.

Results on CIFAR-10 & 100. SNSCL plays against with negative effects of label noise by enhancing distinguishable representation, which is also suitable for generic noisy classification to some degree. We conduct experiments on synthetic noisy CIFAR-10 & 100 and show the results in Appx. C.2. Overall, the performance of tested methods achieve non-trivial improvements by combining with SNSCL.

5.3. More analysis

Effectiveness. Our algorithm exhibits the superior effectiveness in two aspects. 1) we plot the curve of test accuracy in Fig. 4(a). It is clear that the accuracy of CE rises dramatically to a peak and gradually decreases, indicating overfitting to noise under fine-grained datasets. For SNSCL, the testing curve is relatively stable and results in good generalization performance. 2) we test the noise ratios with a wide range of $r \in \{10\%, \cdots, 80\%\}$ and record the best and the last top-1 testing accuracy. As shown in the scatter Fig. 4(b), SNSCL can mitigate reasonable discriminability for a high noise ratio (68% top-1 accuracy for symmetric 80% label noise). Meanwhile, more train curves with varying noise ratios are shown in Appx. C.4.

Sensibility. We explore the effect of two essential hyper-parameters in our method. 1) **The momentum queue size** D. The batch size or momentum queue size is the key point in contrastive learning, and thus we set $\mathrm{D} \in \{4, 8, 16, 32, 48, 64\}$ to explore its influence on our framework. The results are shown in Fig. 4(c). As the size D reaches a certain amount, the performance will not increase. Thus, we set a suitable yet effective value $\mathrm{D} = 32$. 2) **The reliability threshold** t. This threshold in the weight-aware mechanism deeply affects the subsequent two weighted strategies. We adjust its value from the space $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ and plot the results in Fig. 4(d). The best performance is attained on two conditions when $t = 0.5$. Therefore, we set the reliability threshold as 0.5.

Compared with contrastive-based LNL methods. We conduct experiments to compare our method (DivideMix + SNSCL) with MoPro [19] and Sel-CL+ [20], two LNL methods based on contrastive learning. Detailed discus-

Table 4. Compared with previous contrastive-based methods on four noisy benchmarks.

Dataset	CIFAR-10			CIFAR-100			Stanford Dogs		CUB-200-2011	
Noise Type	S. 50%	S. 80%	A. 40%	S. 50%	S. 80%	A. 40%	S. 40%	A. 30%	S. 40%	A. 30%
MoPro [19]	**95.6**	90.1	93.0	**74.9**	61.9	73.0	78.41±0.1	74.39±0.2	73.23±0.2	68.58±0.4
Sel-CL+ [20]	93.9	89.2	93.4	72.4	59.6	74.2	77.92±0.3	75.29±0.2	73.01±0.1	70.47±0.1
Ours	95.2	**91.7**	**94.9**	74.7	**64.3**	**75.1**	**79.13±0.2**	**77.20±0.1**	**73.67±0.3**	**71.28±0.2**

Table 5. Compared stochastic module (Ours) with weak (W.) and strong (S.) augmentation under 40% symmetric label noise.

Strategies	Stanford Dogs	CUB-200-2011	Aircraft
W. aug.	73.24±0.2	66.48±0.2	68.74±0.5
S. aug.	74.02±0.3	67.26±0.4	70.19±0.2
Ours	**75.27±0.2**	69.09±0.4	**70.48±0.3**
Ours + S. aug.	75.13±0.2	**69.31±0.2**	70.19±0.3

sions about these methods can be found in Related works.

Tab. 4 reports the comparison results with top-1 test accuracy on four benchmarks. Our method outperforms Sel-CL+ and MoPro in most noisy settings. As the noise ratio arises, the achievements of SNSCL are more remarkable. Compared to Sel-CL+ while the performance is improved by 2.5% on CIFAR-10 80% symmetric noise, and 4.7% on CIFAR-100 80% symmetric noise. Under four LNL-FG settings, our methods consistently outperform other methods. Compared to Sel-CL+, the improvement is roughly 2% on Stanford Dogs 30% asymmetric noise.

Discussion about the stochastic module. We conduct comparison experiments with traditional augmentation strategies to verify the ability of representation enhancement of stochastic module. As shown in Tab. 5, our proposed module exhibits greater performance under noisy conditions. Compared to strong augmentation, the average improvement is more than 1%. Besides, the combination of our stochastic module and strong augmentation does not bring improvements. Thus, we do not adopt strong augmentation strategies in our training framework.

Ablation study. In our proposed SNSCL, there mainly exists three components, weighted-correction and weighted-update strategy in a weighted-aware mechanism and a stochastic module. We conduct the ablation study on two benchmarks to evaluate the effectiveness of each component and show the results in Tab. 6. Under the settings of Stanford dogs with 40% symmetric noisy labels, the combination of three components improves the performance of CE by more than 6% and the effect of DivideMix by 3% respectively, while all components bring some positive effects. Meanwhile, due to the noise-sensitivity of SCL, integrating SCL into CE brings performance degradation instead. To some extent, these results demonstrate the effectiveness of each part of our method.

Visualization. To demonstrate the distinguishable classes are learnt by our proposed SNSCL, we leverage t-SNE [44] to visualize the feature embeddings on the testing sets of

Table 6. **Ablation study** about the effectiveness of each component under 40% *symm.* label noise.

	Stanford Dogs	CUB-200-2011
CE	69.20 (50.45)	54.14 (45.85)
CE + SCL	68.49 (54.77)	53.30 (45.92)
CE + SNSCL	**75.27 (75.00)**	**68.83 (68.67)**
w/o Weight corr.	70.91±0.6	62.71±0.5
w/o Weight update	73.45±0.3	65.29±0.4
w/o Stoc. module	74.11±0.3	67.44±0.3
DivideMix	77.93 (76.28)	67.35 (66.96)
DivideMix + SCL	78.20 (77.89)	70.28 (70.02)
DivideMix + SNSCL	**79.12 (78.91)**	**73.66 (73.28)**
w/o Weight corr.	78.30±0.2	70.41±0.3
w/o Weight update	78.52±0.1	72.59±0.2
w/o Stoc. module	78.85±0.1	73.06±0.1

CIFAR-10 & CIFAR-100. The results are shown in Appx. C.3, verifying the improvement of SNSCL on representation learning under noisy conditions.

6. Conclusion

In this work, we propose a novel task called LNL-FG, posing a more challenging noisy scenario to learning with noisy labels. For this, we design a general framework called SNSCL. SNSCL contains a noise-tolerated contrastive loss and a stochastic module. Compared with typical SCL, our contrastive learning framework incorporates a weight-aware mechanism which corrects noisy labels and selectively update momentum queue lists. Besides, we propose a stochastic module for feature transformation, generating the probabilistic distribution of feature embeddings. We achieve greater representation ability by sampling transformed embedding from this distribution. SNSCL is applicable to prevailing LNL methods and further improves their generalization performance on LNL-FG.

Acknowledgements

Haoliang Sun was supported by Natural Science Foundation of China (No. 62106129), China Postdoctoral Science Foundation (No. 2021TQ0195, 2021M701984) and Natural Science Foundation of Shandong Province (No. ZR2021QF053). Yilong Yin was supported by Major Basic Research Project of Natural Science Foundation of Shandong Province (No. ZR2021ZD15) and Natural Science Foundation of China (No. 62176139). Lei Feng was supported by the National Natural Science Foundation of China (No. 62106028), Chongqing Overseas Chinese Entrepreneurship and Innovation Support Program, and CAAI-Huawei MindSpore Open Fund.

References

[1] Yingbin Bai and Tongliang Liu. Me-momentum: Extracting hard confident examples from noisily labeled data. In *ICCV*, 2021. 1, 2

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, 2014. 5

[3] Guy Bukchin, Eli Schwartz, Kate Saenko, Ori Shahar, Rogerio Feris, Raja Giryes, and Leonid Karlinsky. Fine-grained angular contrastive learning with coarse labels. In *CVPR*, 2021. 2

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2, 3

[5] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020. 3

[6] Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. In *ICLR*, 2021. 1

[7] Hao Cheng, Zhaowei Zhu, Xing Sun, and Yang Liu. Mitigating memorization of noisy labels via regularization between representations. In *ICLR*, 2023. 1

[8] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, 2020. 2, 3

[9] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018. 1, 2, 6

[10] Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *ICCV*, 2019. 7

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2, 3, 4

[12] Rundong He, Zhongyi Han, Xiankai Lu, and Yilong Yin. Safe-student for safe deep semi-supervised learning with unseen-class unlabeled data. In *CVPR*, 2022. 1

[13] Rundong He, Zhongyi Han, Yang Yang, and Yilong Yin. Not all parameters should be treated equally: Deep safe semi-supervised learning under class distribution mismatch. In *AAAI*, 2022. 1

[14] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9, 2020. 2

[15] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020. 2, 3

[16] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *NeurIPS*, 2015. 4

[17] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *CVPR*, 2018. 7

[18] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *ICLR*, 2020. 1, 6, 7

[19] Junnan Li, Caiming Xiong, and Steven CH Hoi. Mopro: Webly supervised learning with momentum prototypes. In *ICLR*, 2021. 3, 7, 8

[20] Shikun Li, Xiaobo Xia, Shiming Ge, and Tongliang Liu. Selective-supervised contrastive learning with noisy labels. In *CVPR*, 2022. 3, 7, 8

[21] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *NeurIPS*, 2020. 1, 7

[22] Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *ICML*, 2020. 2

[23] Philip M Long and Rocco A Servedio. Random classification noise defeats all convex potential boosters. In *ICML*, 2008. 1

[24] Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *ICML*, 2020. 6

[25] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *ICML*, 2020. 2

[26] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *NeurIPS*, 2013. 1

[27] Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. Self: Learning to filter noisy labels with self-ensembling. In *ICLR*, 2020. 2

[28] Xiushan Nie, Weizhen Jing, Chaoran Cui, Chen Jason Zhang, Lei Zhu, and Yilong Yin. Joint multi-view hashing for large-scale near-duplicate video retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 32(10):1951–1965, 2019. 1

[29] Xiushan Nie, Xingbo Liu, Jie Guo, Letian Wang, and Yilong Yin. Supervised discrete multiple-length hashing for image retrieval. *IEEE Transactions on Big Data*, 2023. 1

[30] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv*, 2018. 3

[31] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *ECCV*, 2020. 2

[32] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, 2017. 7

[33] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. In *ICLR*, 2017. 6

[34] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, 2018. 3

[35] Karishma Sharma, Pinar Donmez, Enming Luo, Yan Liu, and I Zeki Yalniz. Noiserank: Unsupervised label noise reduction with dependence models. In *ECCV*, 2020. 7

[36] Yang Shi, Xiushan Nie, Xingbo Liu, Lu Yang, and Yilong Yin. Zero-shot hashing via asymmetric ratio similarity matrix. *IEEE Transactions on Knowledge and Data Engineering*, 2022. 1

[37] Yang Shi, Xiushan Nie, Xingbo Liu, Li Zou, and Yilong Yin. Supervised adaptive similarity matrix hashing. *IEEE Transactions on Image Processing*, 2022. 1

[38] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *NeurIPS*, 2019. 1, 3, 6, 7

[39] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *ICML*, 2019. 2

[40] Haoliang Sun, Chenhui Guo, Qi Wei, Zhongyi Han, and Yilong Yin. Learning to rectify for robust learning with noisy labels. *Pattern Recognition*, 2022. 1, 7

[41] Haoliang Sun, Xiantong Zhen, Chris Bailey, Parham Rasoulinejad, Yilong Yin, and Shuo Li. Direct estimation of spinal cobb angles by structured multi-output regression. In *ICIP*, 2017. 1

[42] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, 2018. 1, 7

[43] Ambuj Tewari and Peter L Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 2007. 1

[44] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008. 8

[45] Vikas Verma, Thang Luong, Kenji Kawaguchi, Hieu Pham, and Quoc Le. Towards domain-agnostic contrastive learning. In *ICML*, 2021. 4

[46] Fan Wang, Zhongyi Han, Yongshun Gong, and Yilong Yin. Exploring domain-invariant parameters for source free domain adaptation. In *CVPR*, 2022. 1

[47] Fan Wang, Zhongyi Han, Zhiyan Zhang, and Yilong Yin. Active source free domain adaptation. *arXiv*, 2022. 1

[48] Ren Wang, Haoliang Sun, Yuling Ma, Xiaoming Xi, and Yilong Yin. Metaviewer: Towards a unified multi-view representation. *arXiv*, 2023. 1

[49] Ren Wang, Haoliang Sun, Xiushan Nie, and Yilong Yin. Snip-fsl: Finding task-specific lottery jackpots for few-shot learning. *Knowledge-Based Systems*, 247:108427, 2022. 1

[50] Ximei Wang, Jinghan Gao, Mingsheng Long, and Jianmin Wang. Self-tuning for data-efficient deep learning. In *ICML*, 2021. 3

[51] Xinshao Wang, Yang Hua, Elyor Kodirov, David A Clifton, and Neil M Robertson. Proselflc: Progressive self label correction for training robust deep neural networks. In *CVPR*, 2021. 2

[52] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *ICCV*, 2019. 2, 6, 7

[53] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *CVPR*, 2020. 1, 2, 6, 7

[54] Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. In *arXiv:2110.12088*, 2021. 1

[55] Qi Wei, Haoliang Sun, Xiankai Lu, and Yilong Yin. Self-filtering: A noise-aware sample selection for label noise with confidence penalization. In *ECCV*, 2022. 1, 2, 5, 7

[56] Yichen Wu, Jun Shu, Qi Xie, Qian Zhao, and Deyu Meng. Learning to purify noisy labels via meta soft label corrector. In *AAAI*, 2021. 2

[57] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, 2015. 5

[58] Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_dmi: An information-theoretic noise-robust loss function. In *NeurIPS*, 2019. 1

[59] Yivan Zhang, Gang Niu, and Masashi Sugiyama. Learning noise transition matrix from only noisy labels via total variation regularization. In *ICML*, 2021. 7

[60] Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, 2018. 2, 6

[61] Guoqing Zheng, Ahmed Hassan Awadallah, and Susan Dumais. Meta label correction for noisy label learning. In *AAAI*, 2021. 2, 6